

MATLAB EXPO 2018

Speeding Up Simulation

Sonia Bridge



Why Speed up Simulation?

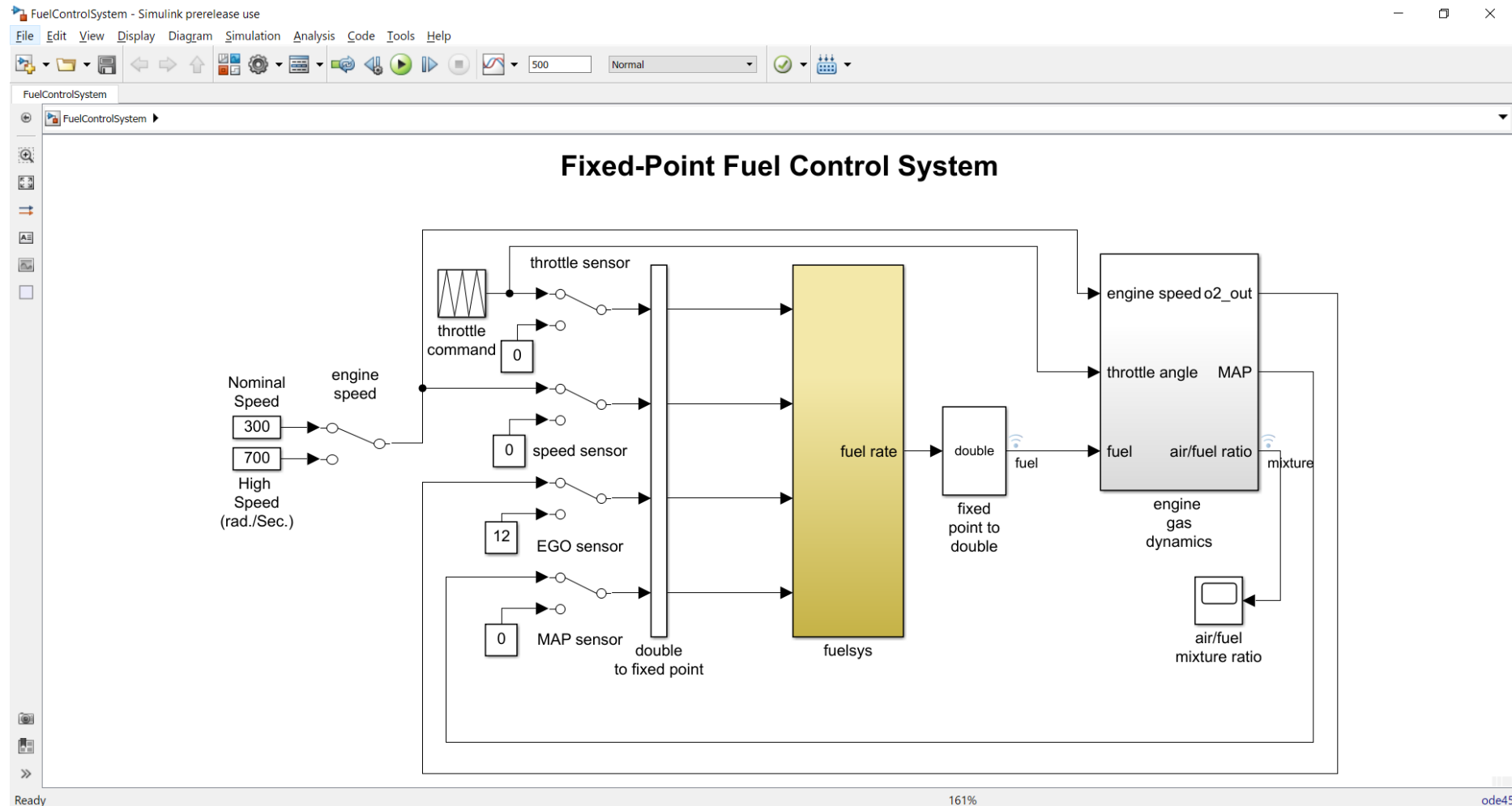
- “I have a **big model** and need to run **long simulations** as quickly as possible”
- “I need to perform **design optimization as quickly as possible**, which requires running **many simulations** as my design parameters change”
- “I need to perform **thousands of Monte Carlo simulations** as quickly as possible”
- “My model **takes forever to update**”



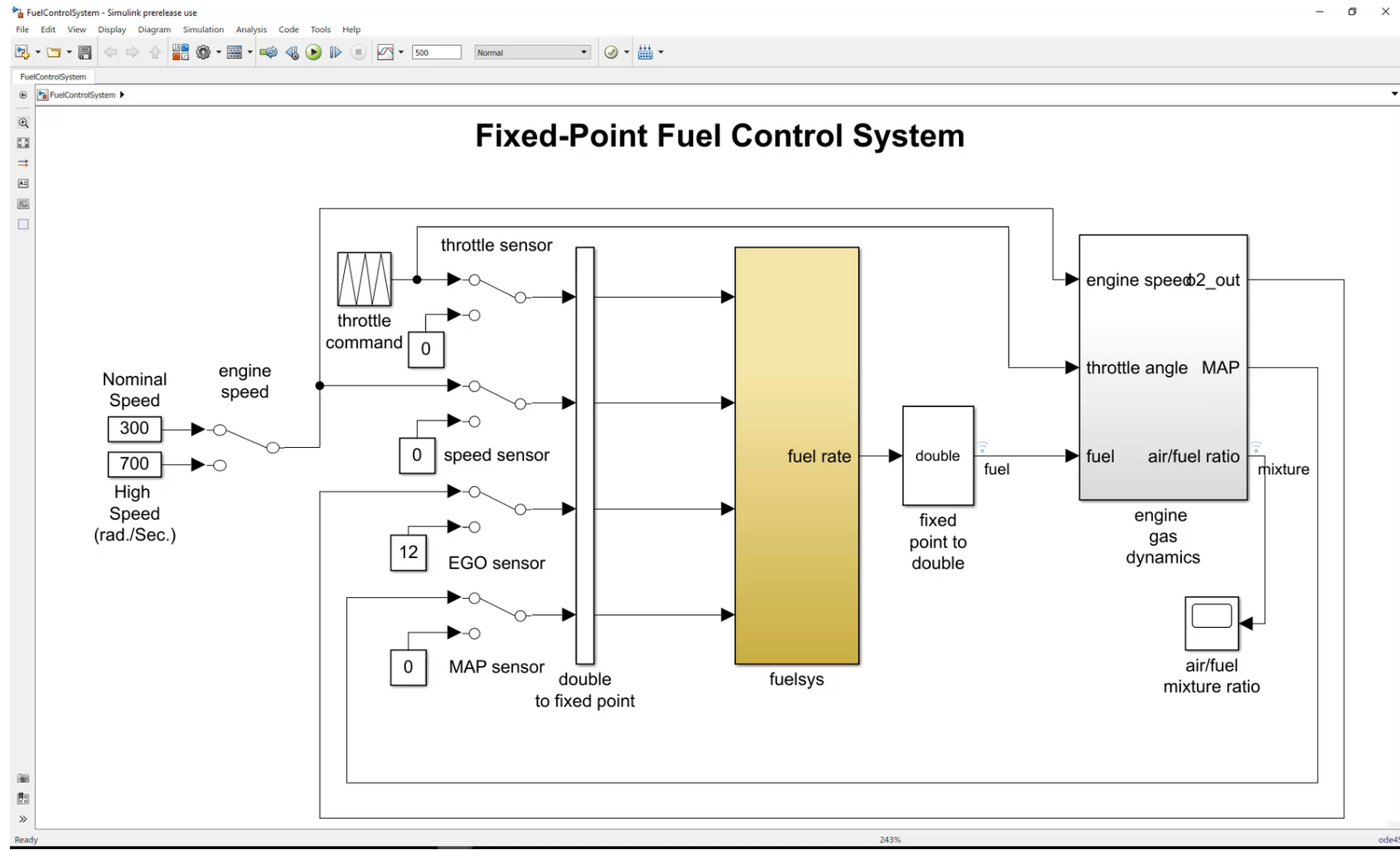
Agenda

- Acceleration mode
- Performance Advisor
- Fast Restart, **parsim** and **batchsim**
- Incremental workflows with Model Referencing

Let's look at an example model to see how Acceleration can speed up simulations



Normal Mode



Agenda

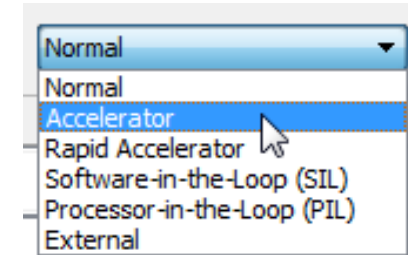
- Acceleration mode
- Performance Advisor
- Fast Restart, `parsim` and `batchsim`
- Incremental workflows using Model Referencing



Accelerator Mode

Why would Simulink speed up?

- JIT compiles (or generates C-code for) portions of the model
- Running compiled code has less overhead



What's the tradeoff?

- There is an overhead for generating the acceleration target
- Some runtime diagnostics are disabled, e.g., inf/nan checking
- May not speed up all models

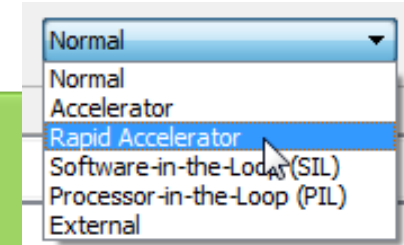
Introduced before **R2006a**



Use Rapid Accelerator Mode

Why would Simulink speed up?

- The Rapid Accelerator mode creates and runs a standalone executable from the model, which has little overhead.
- If possible, this executable runs on a separate core than the MATLAB session



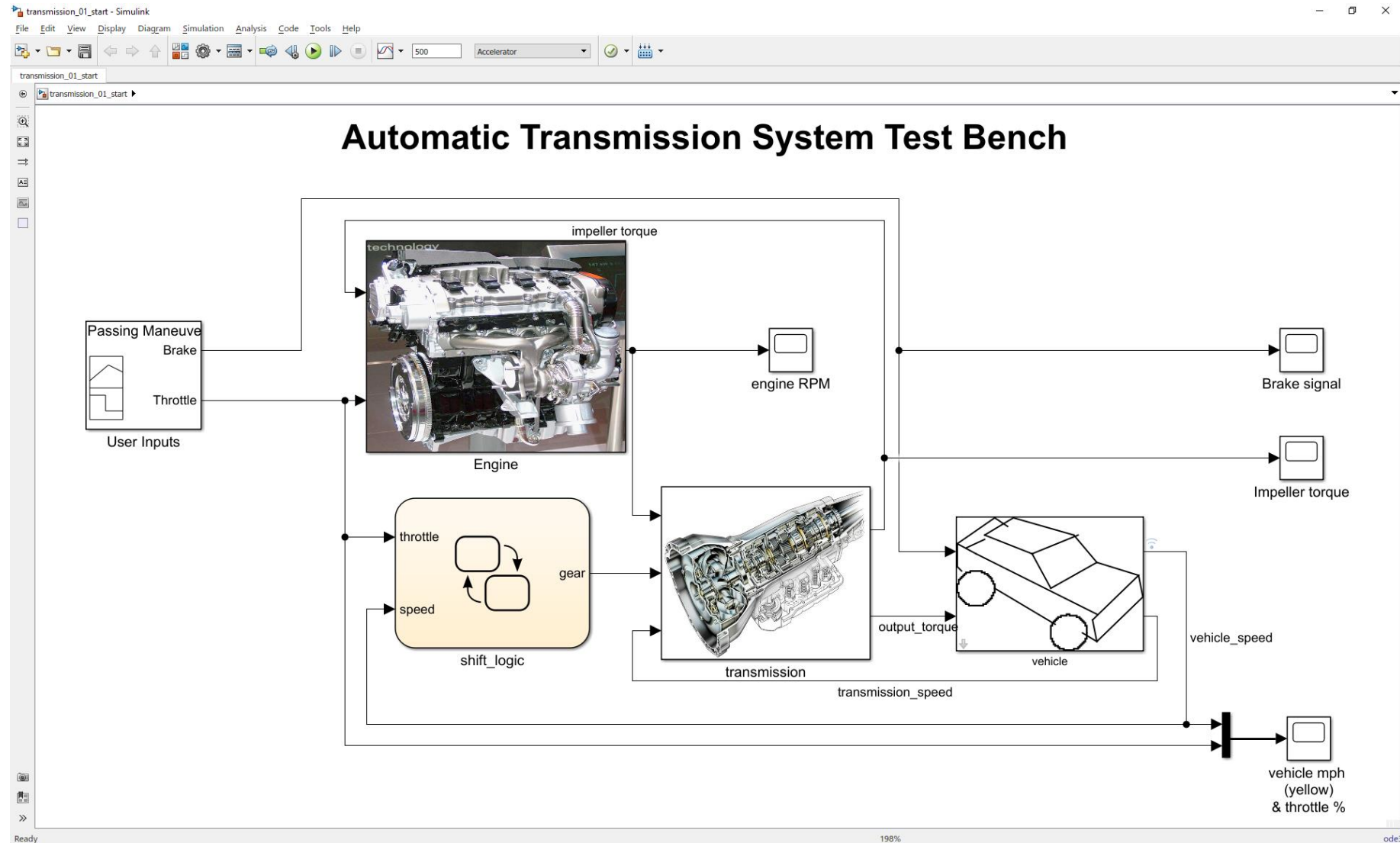
What's the tradeoff?

- It takes time to build the Rapid Acceleration target
- Debugging capabilities are disabled, except for scopes and viewers
- Entire model needs to support code generation

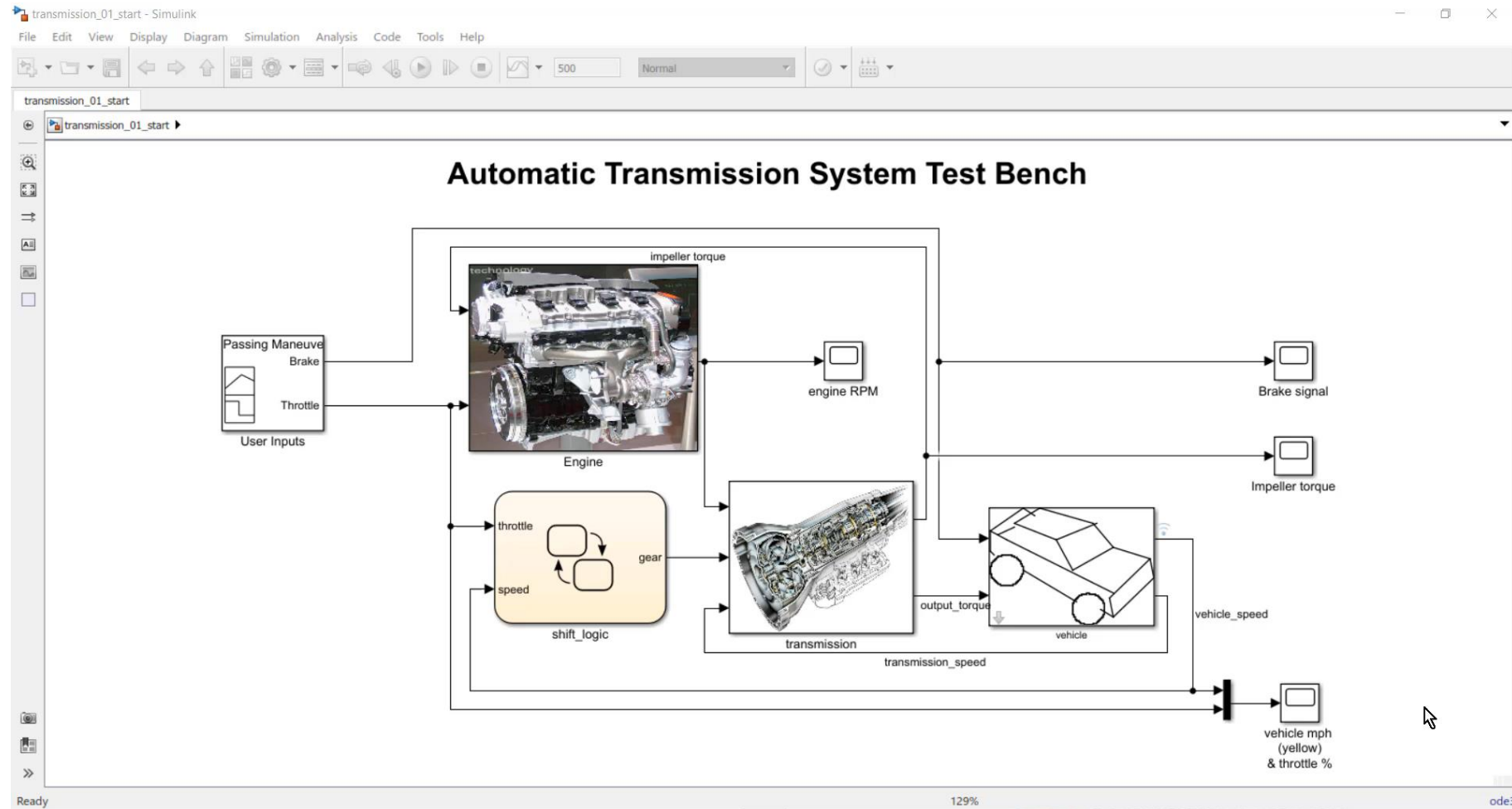
Agenda

- Acceleration mode
- Performance Advisor
- Fast Restart, `parsim` and `batchsim`
- Incremental workflows using Model Referencing

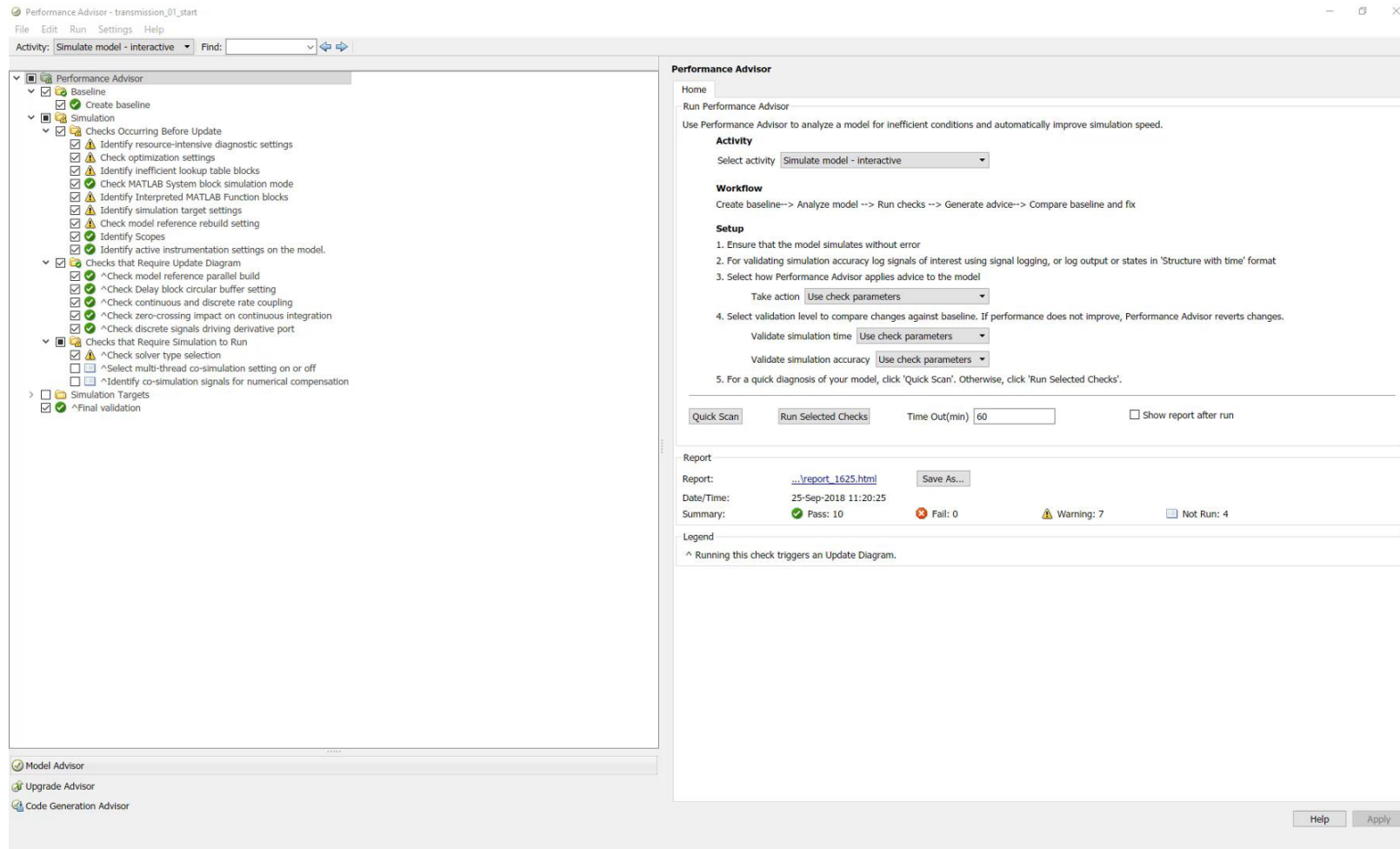
Performance Advisor Example



Let's run the Performance Advisor on the model



Performance Advisor – Results



The screenshot displays the Performance Advisor window for a model named 'transmission_01_start'. The interface is divided into two main panes. The left pane shows a tree view of various checks categorized under 'Performance Advisor', 'Baseline', 'Simulation', and 'Simulation Targets'. The right pane provides a detailed view of the 'Performance Advisor' settings and results.

Performance Advisor Settings:

- Activity:** Simulate model - interactive
- Workflow:** Create baseline--> Analyze model --> Run checks --> Generate advice--> Compare baseline and fix
- Setup:**
 - Ensure that the model simulates without error
 - For validating simulation accuracy log signals of interest using signal logging, or log output or states in 'Structure with time' format
 - Select how Performance Advisor applies advice to the model
 - Take action: Use check parameters
 - Select validation level to compare changes against baseline. If performance does not improve, Performance Advisor reverts changes.
 - Validate simulation time: Use check parameters
 - Validate simulation accuracy: Use check parameters
 - For a quick diagnosis of your model, click 'Quick Scan'. Otherwise, click 'Run Selected Checks'.

Buttons: Quick Scan, Run Selected Checks, Time Out(min): 60, Show report after run

Report:

Report: [..report_1625.html](#) Save As...

Date/Time: 25-Sep-2018 11:20:25

Summary: ✔ Pass: 10 ✖ Fail: 0 ⚠ Warning: 7 📄 Not Run: 4

Legend:

^ Running this check triggers an Update Diagram.

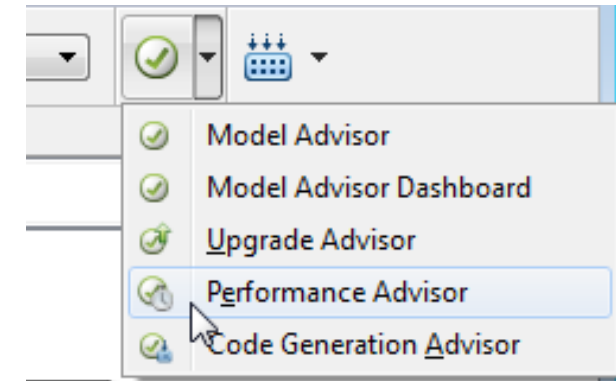
Performance Advisor

Why would Simulink speed up?

- Checks your model for speedup options
- Validates its own advice, only applies changes that:
 - Give the same answer
 - Improve speed

What's the tradeoff?

- Takes time to run the analysis
- Not comprehensive
 - Trading off fidelity for speed is not part of performance advisor



Agenda

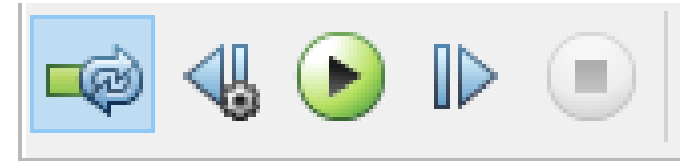
- Acceleration mode
- Performance Advisor
- Fast Restart, **parsim** and **batchsim**
- Incremental workflows using Model Referencing



Fast Restart

Why would Simulink speed up?

- Avoids recompilation between simulation runs
- Works with Accelerator mode



What's the tradeoff?

- Cannot edit the model when in fast restart mode

Running Multiple Simulations in Parallel

- Use `SimulationInput` object to specify changes to a model for simulations
- Use `parsim` to run parallel simulations
- Use **Simulation Manager** to monitor and inspect results from multiple simulations

R2017a

R2017a

R2017b

Benefits of using `parsim`

```
out = parsim(in) ;
```

That's it!

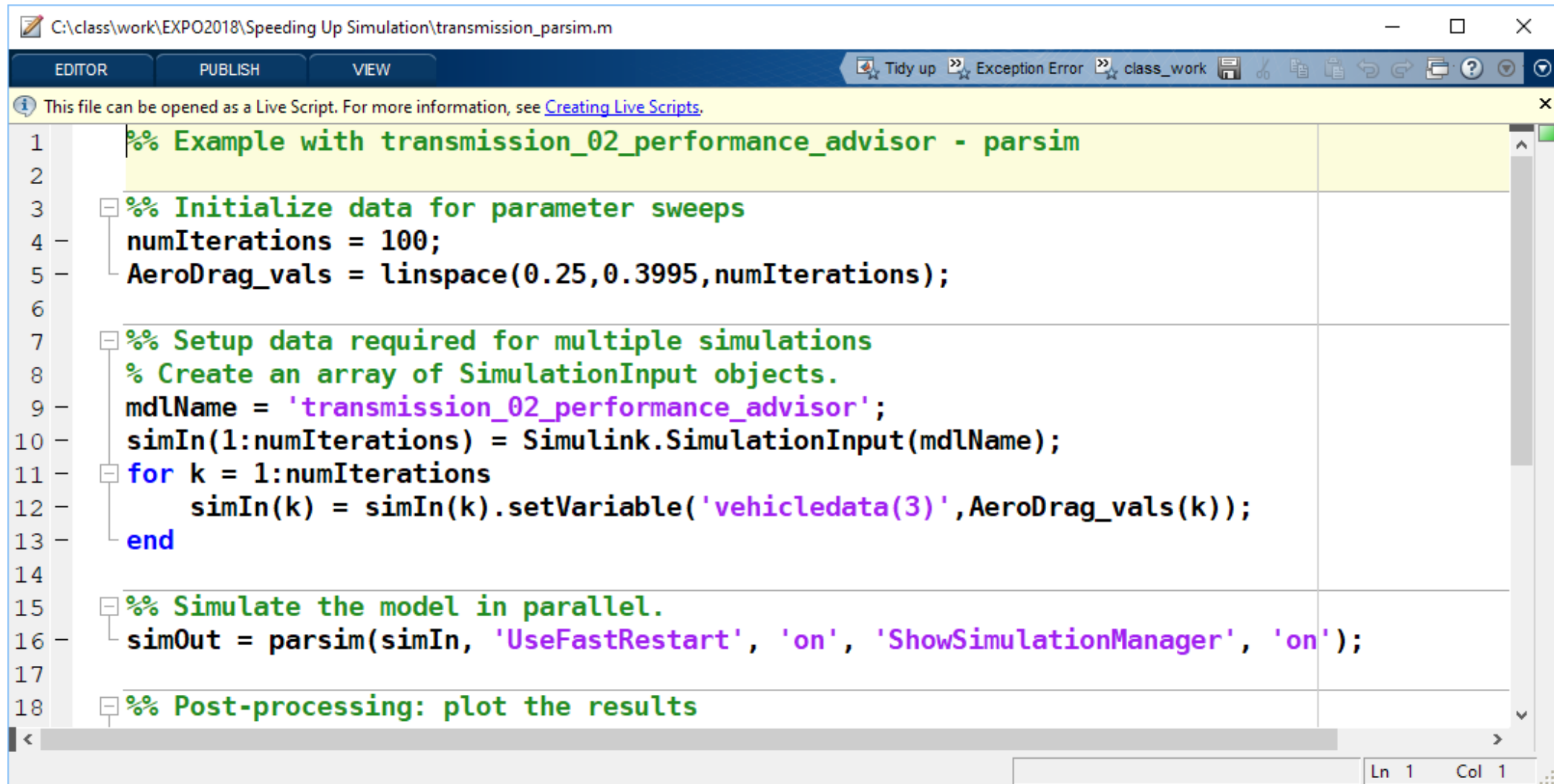
One command to run them all!

- **parsim** “mistake-proofs” your workflow
 - Handle model dependencies
 - Automatic management of build folders
 - Parallel builds of model references
 - Transfer base workspace variables to workers
 - Error diagnostics
 - Progress display
 - And more ...

`parsim` manages
the details of
running parallel
simulations

... so you can focus
on the design tasks!

Code example for parameter sweep using parsim

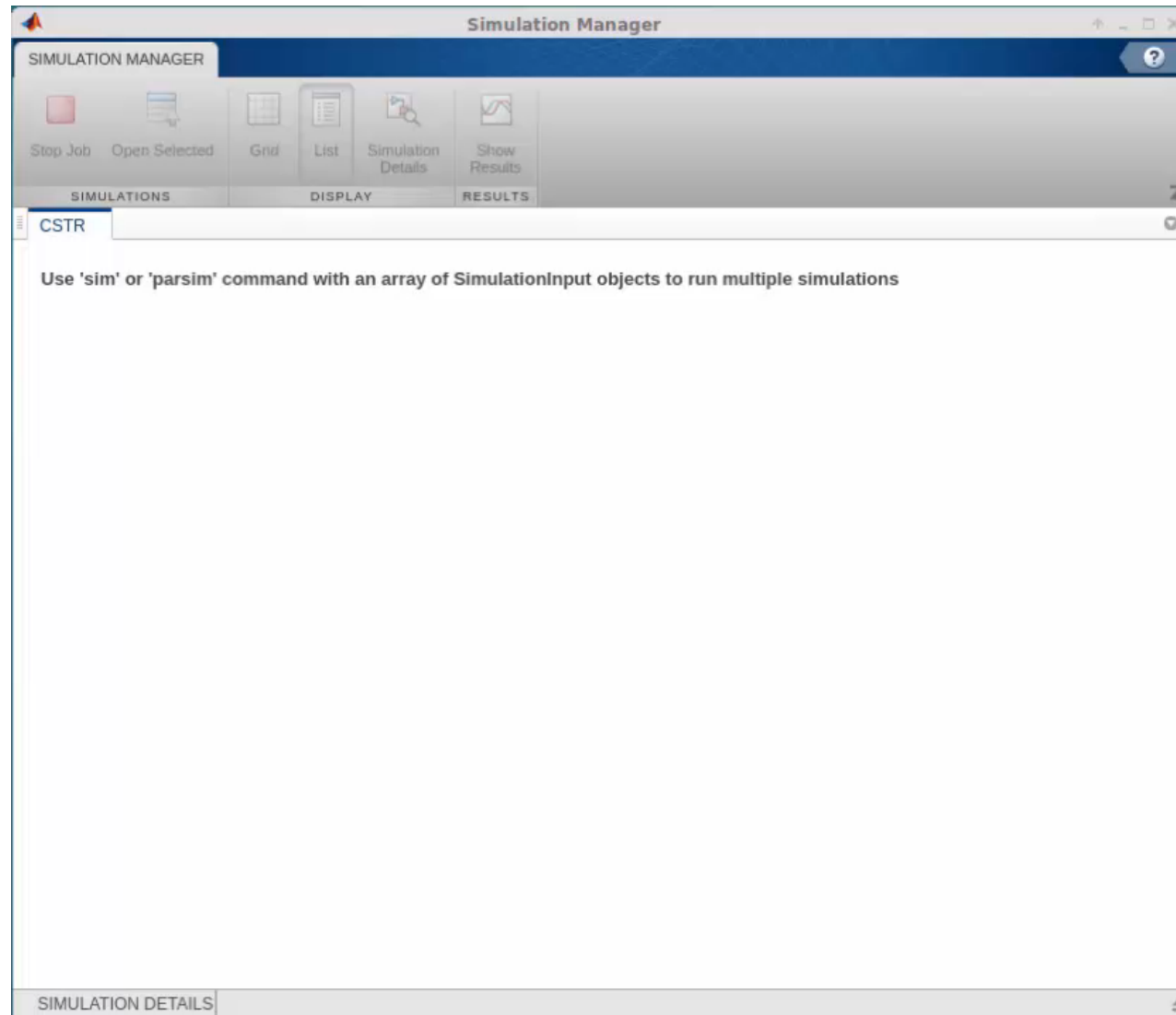


The screenshot shows the MATLAB Editor window with the file `C:\class\work\EXPO2018\Speeding Up Simulation\transmission_parsim.m` open. The editor has tabs for EDITOR, PUBLISH, and VIEW. A toolbar at the top includes icons for Tidy up, Exception Error, and class_work. A message bar indicates the file can be opened as a Live Script. The script content is as follows:

```
1 %% Example with transmission_02_performance_advisor - parsim
2
3 %% Initialize data for parameter sweeps
4 numIterations = 100;
5 AeroDrag_vals = linspace(0.25,0.3995,numIterations);
6
7 %% Setup data required for multiple simulations
8 % Create an array of SimulationInput objects.
9 mdlName = 'transmission_02_performance_advisor';
10 simIn(1:numIterations) = Simulink.SimulationInput(mdlName);
11 for k = 1:numIterations
12     simIn(k) = simIn(k).setVariable('vehicledata(3)',AeroDrag_vals(k));
13 end
14
15 %% Simulate the model in parallel.
16 simOut = parsim(simIn, 'UseFastRestart', 'on', 'ShowSimulationManager', 'on');
17
18 %% Post-processing: plot the results
```

The status bar at the bottom right shows "Ln 1 Col 1".

Simulation Manager Example



parsim

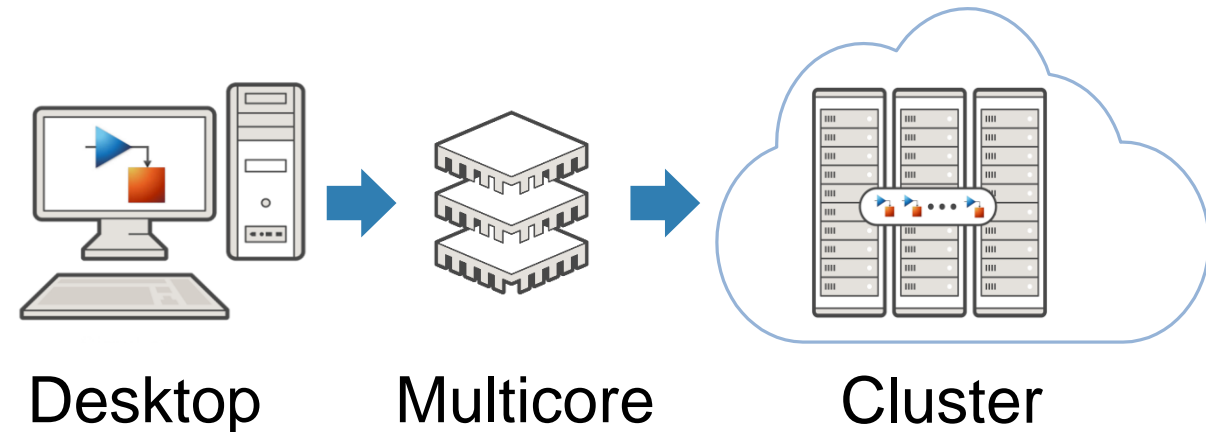
Why would Simulink speed up?

- Runs simulations in parallel using MATLAB Parallel Computing
- Parallelization details are automatically handled

```
for i = 10000:-1:1
    in(i) = Simulink.SimulationInput('my_model');
    in(i) = in(i).setVariable('my_var', i);
end
out = parsim(in);
```

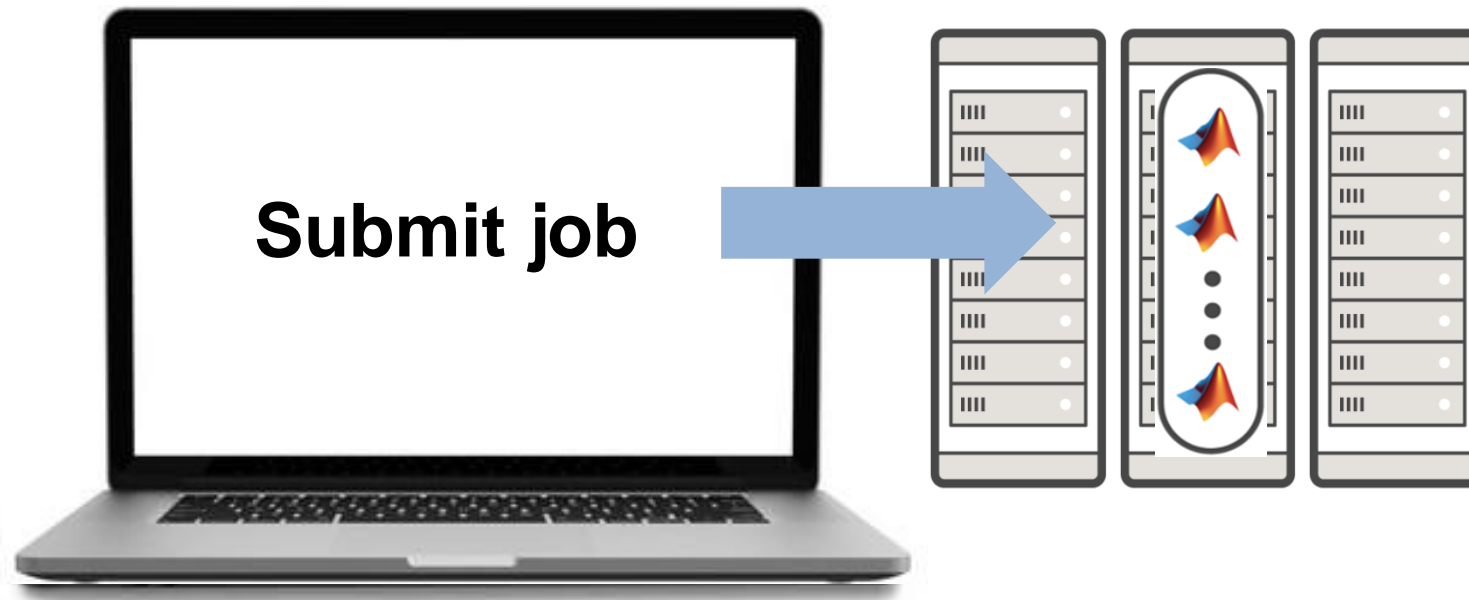
What's the tradeoff?

- Overhead of setting up parallel pool
- Overhead of starting simulations on the workers
- Needs scripting in MATLAB

**R2017a**

Batch Workflows

- Submit jobs and retrieve results later



Batch Workflows

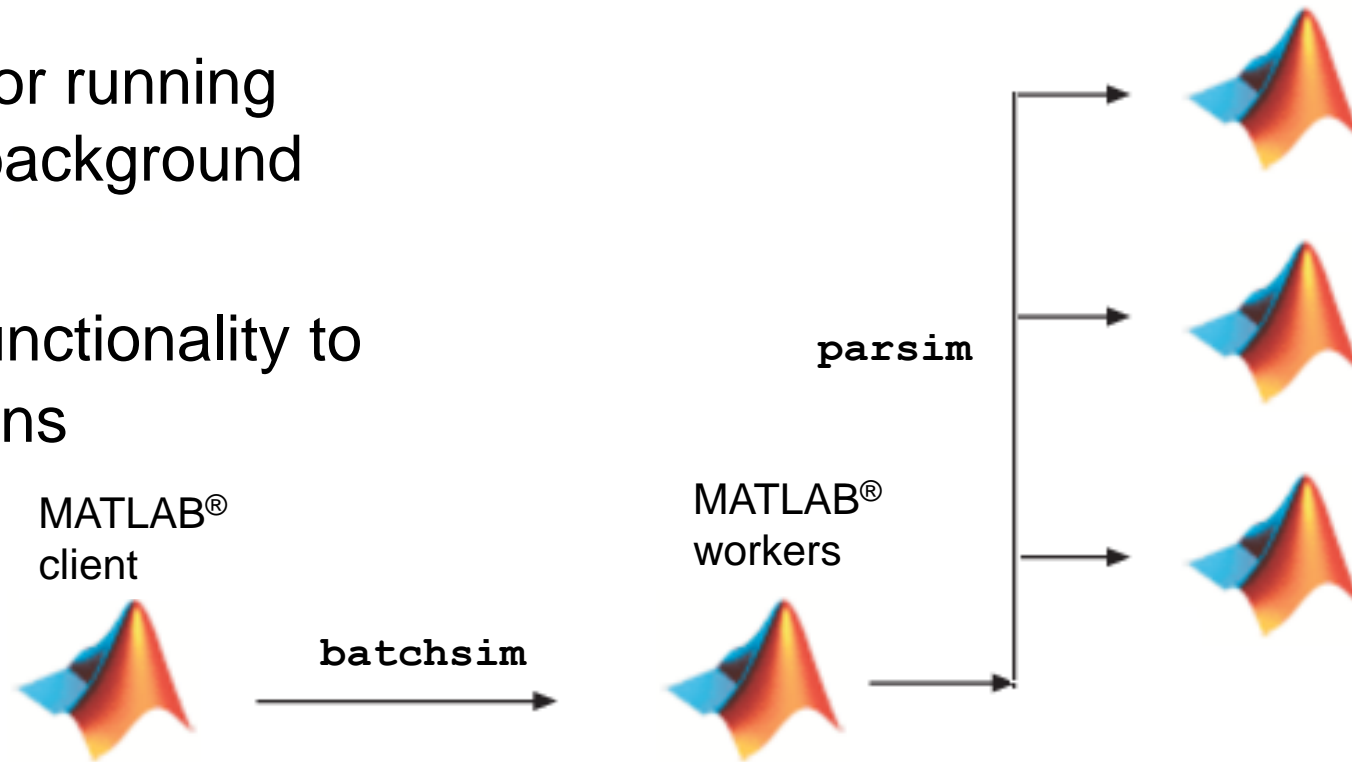
- Submit jobs and retrieve results later



batchsim automates batch workflows

```
job = batchsim(in) ;
```

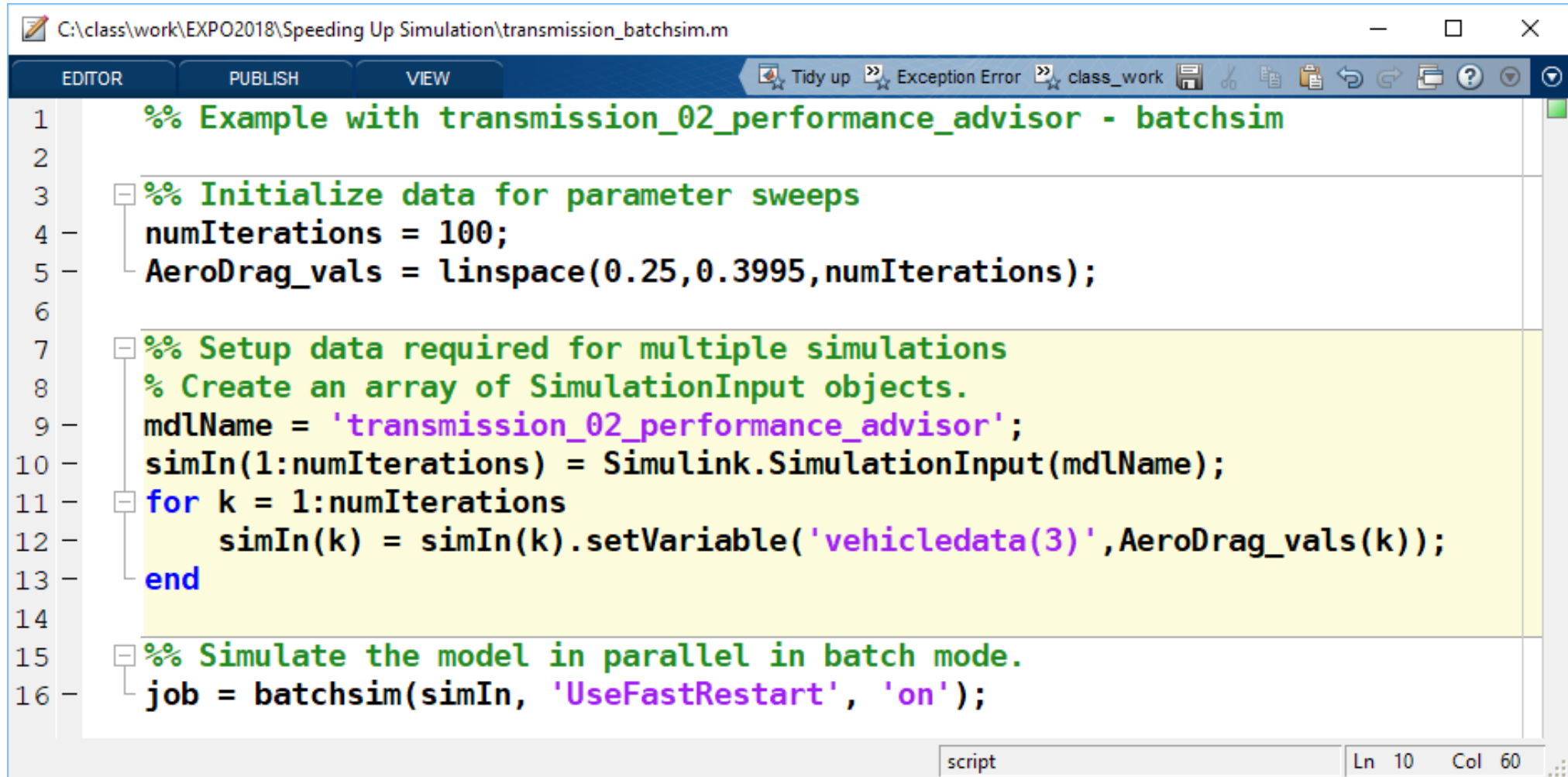
- Automates set up for running simulations in the background
- Extends **parsim** functionality to support batch options

**R2018b**

parsim VS. batchsim

parsim	batchsim
Blocking	Non-blocking
Possible to obtain intermediate results on-the-fly	Simulations are offloaded; retrieve complete results later
Interactive	Not interactive

Simulation code for batchsim



The image shows a MATLAB Editor window titled 'C:\class\work\EXPO2018\Speeding Up Simulation\transmission_batchsim.m'. The window has tabs for EDITOR, PUBLISH, and VIEW. The code is as follows:

```
1 %% Example with transmission_02_performance_advisor - batchsim
2
3 %% Initialize data for parameter sweeps
4 numIterations = 100;
5 AeroDrag_vals = linspace(0.25,0.3995,numIterations);
6
7 %% Setup data required for multiple simulations
8 % Create an array of SimulationInput objects.
9 mdlName = 'transmission_02_performance_advisor';
10 simIn(1:numIterations) = Simulink.SimulationInput(mdlName);
11 for k = 1:numIterations
12     simIn(k) = simIn(k).setVariable('vehicledata(3)',AeroDrag_vals(k));
13 end
14
15 %% Simulate the model in parallel in batch mode.
16 job = batchsim(simIn, 'UseFastRestart', 'on');
```

The status bar at the bottom indicates 'script', 'Ln 10', and 'Col 60'.

Post-processing code for batchsim

CALLBACKS

Function that runs on the client when job reaches the finished state
JobFinishedFcn

@processResults

C:\class\work\EXPO2018\Speeding Up Simulation\processResults.m

EDITOR PUBLISH VIEW Tidy up Exception F class_work

```
1 function processResults(job, eventData)
2 % Get results
3 simOut = fetchOutputs(job); % results are returned in a cell array
4 simOut = simOut{1};
5 % Cleanup
6 delete(job);
7
8 %% Post-processing: plot the results
```

3 usages of "AeroDrag_vals" found processResults Ln 20 Col 38

Agenda

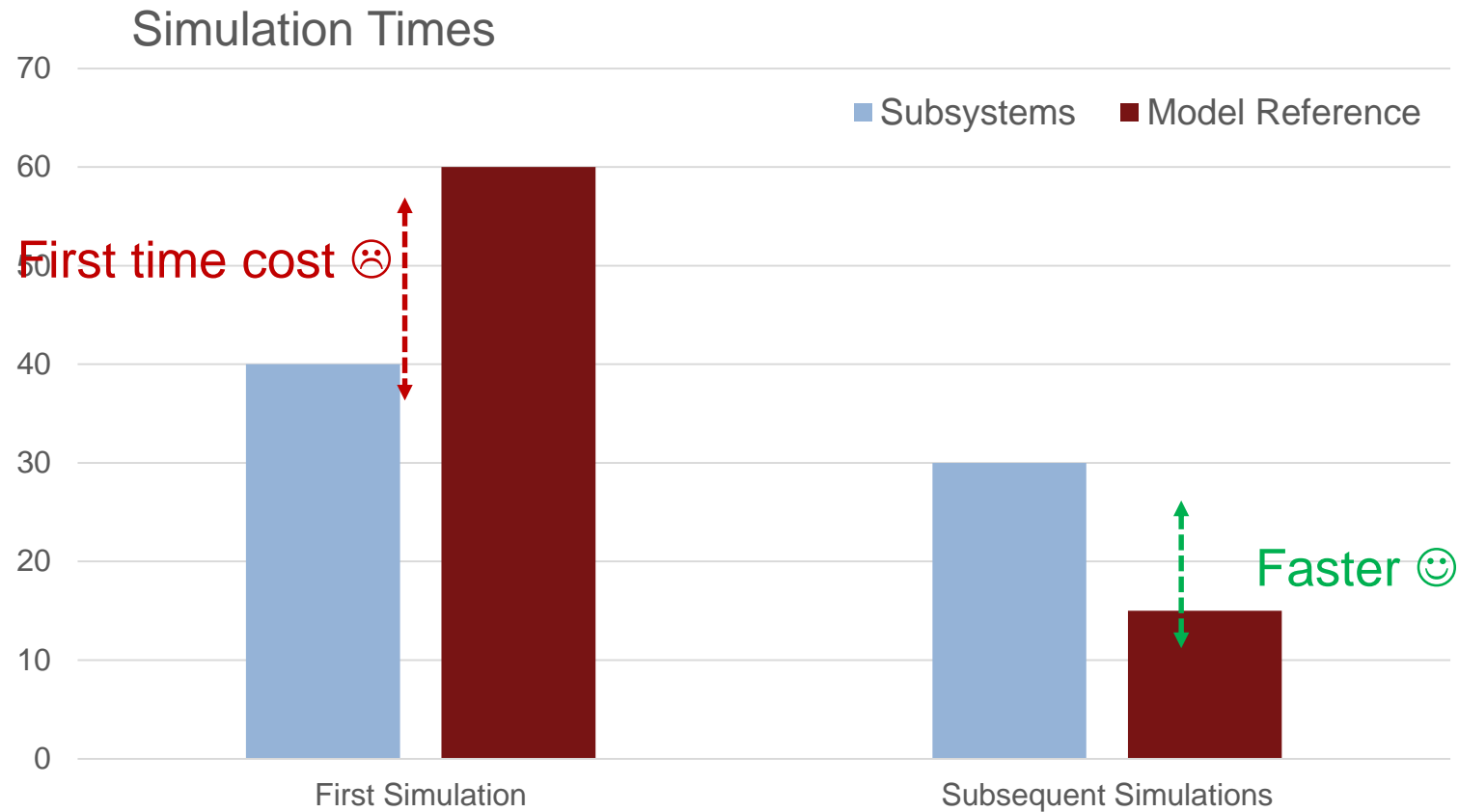
- Acceleration mode
- Performance Advisor
- Fast Restart, `parsim` and `batchsim`
- Incremental workflows using Model Referencing

What is an incremental workflow?

*“My model **takes forever to update**”*

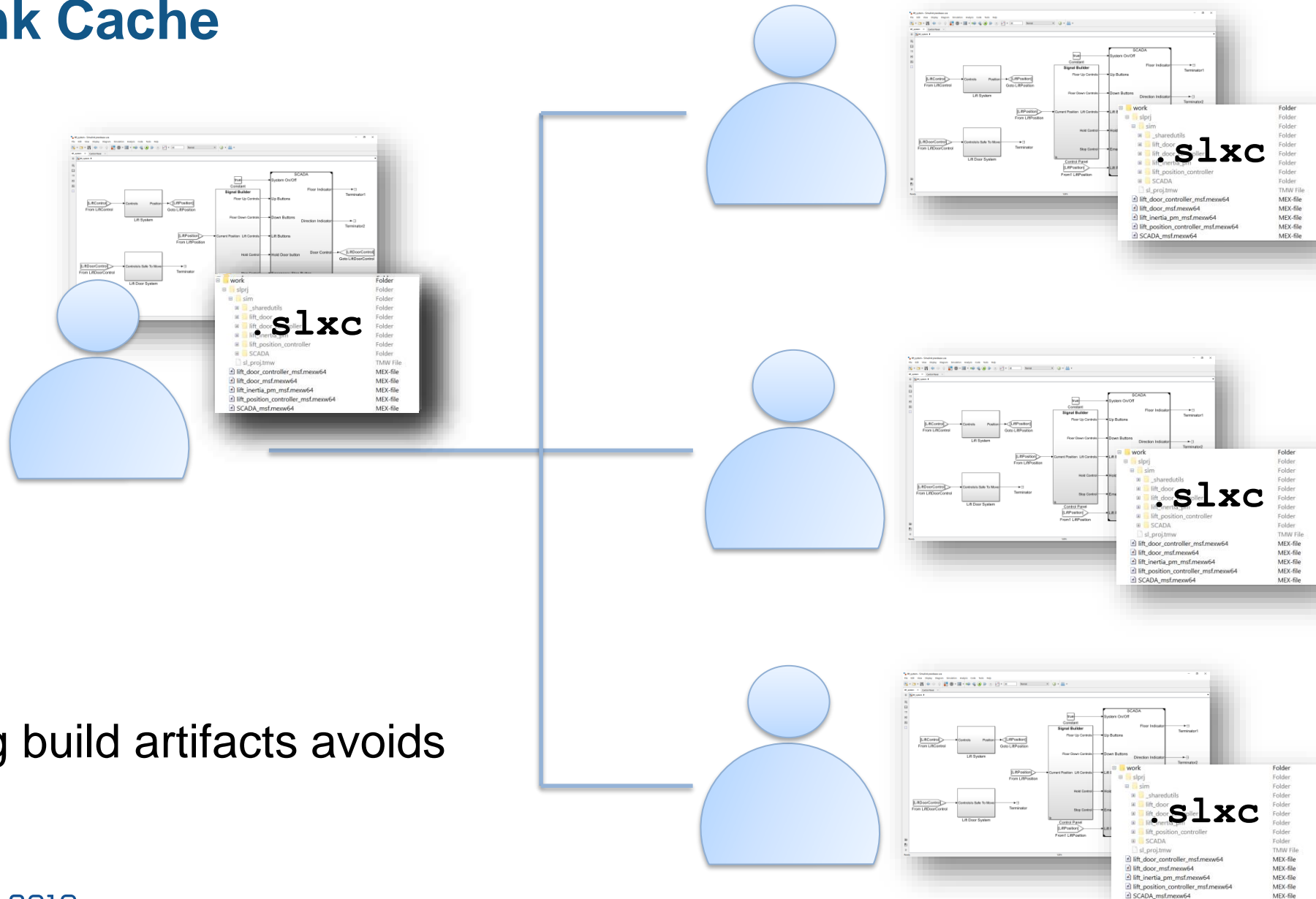
Only perform an action when necessary;
reuse and cache as much as possible

Model Reference: Performance



- Documentation: [Design Partitioning](#)

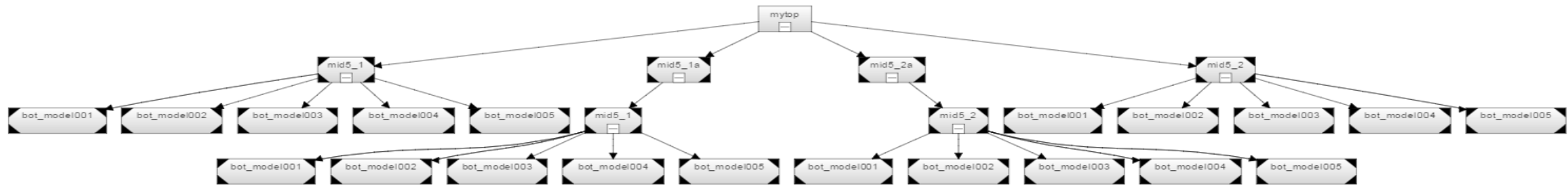
Simulink Cache



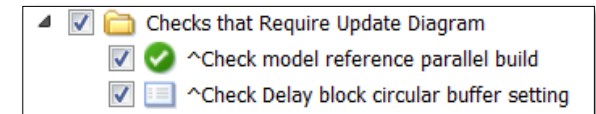
- Sharing build artifacts avoids rebuild

Parallel Builds for Model Reference Hierarchies

- Speed up updates for models with large model reference hierarchies:



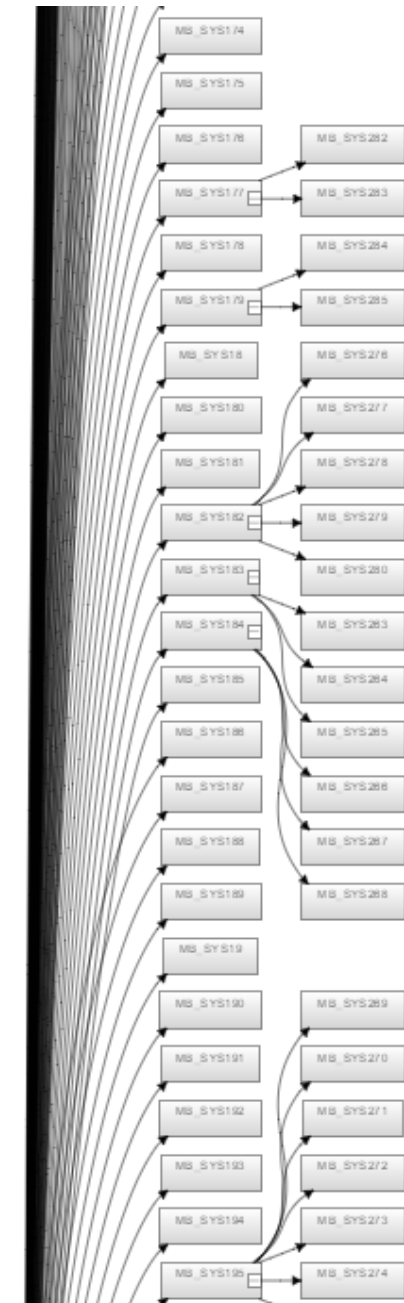
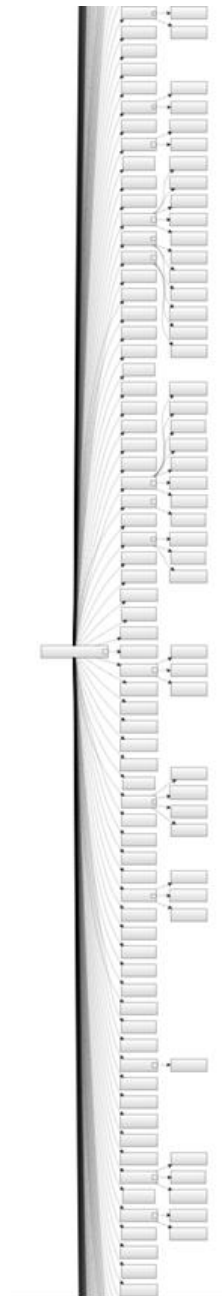
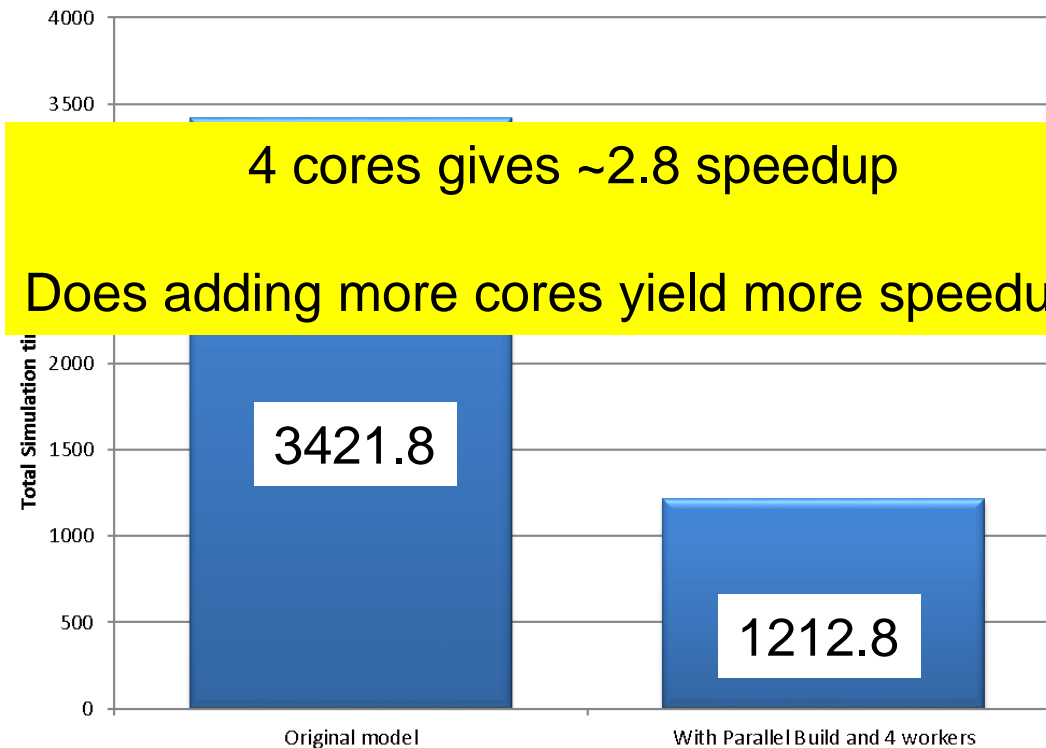
- Enable parallel builds using Configuration Parameters
- Performance Advisor checks for parallel build opportunities
- Enable the **If any changes in known dependencies detected** build option



Model Reference Parallel Build User Example

- Approximately 400 referenced models

Model Update Time comparison of first-time build
with and without PCT



Performance Advisor: Check model reference parallel build

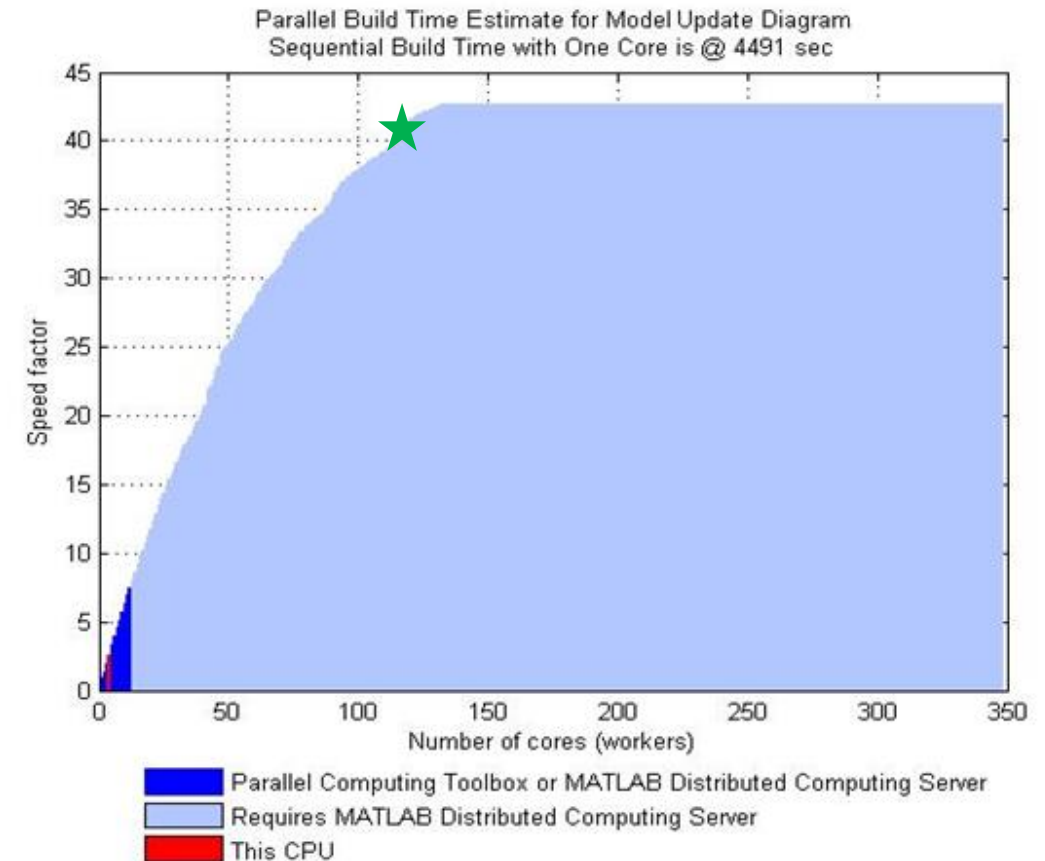
- Performance Advisor estimates the speedup with more cores
- The estimated speed up with 4 cores is ~2.6
 - Close to the measured value ~2.8
- Given ~120 cores, the estimated speed up is ~42
 - => Build time goes from ~3400s to ~80s

Analysis and Advice:

Estimate build time speedup using number of 4 cores of this CPU: ---- --2.6268x

Estimate build time speedup using 348 workers: ---- --42.6475x

Estimated build times for various worker counts are as follows:



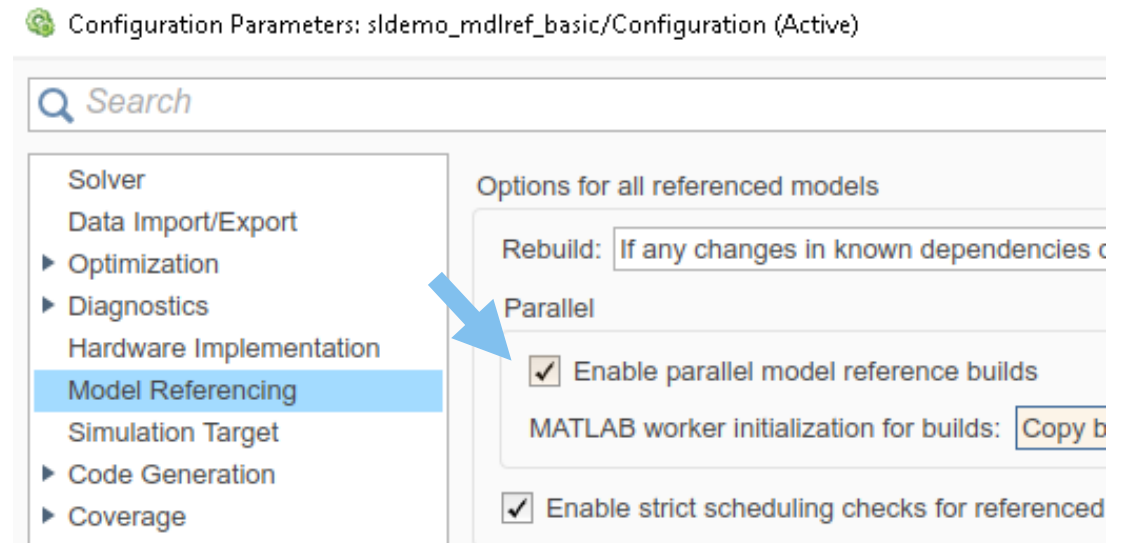
Model Reference Parallel Build

Why would Simulink speed up?

- Model reference targets are built in parallel
- Use Performance Advisor to check if your large models can benefit from this option

What's the tradeoff?

- Speedup is model dependent
- Requires MATLAB Parallel Computing



Learn more about additional speedup tips

- Use of Simulink Profiler and `sldiagnostics` to identify bottlenecks
 - Accelerate the initialization phase
 - Reduce model interactivity
 - Reduce model complexity with alternatives such as look-up tables and linear models
 - Choose and configure a Solver
 - Save the Simulation State
-
- Documentation: [Modeling Techniques that Improve Performance](#)

Key Takeaways

- Recommended steps to easily speed up your Simulink models
- How parallel computing tools decrease the time to run multiple simulations

Q & A